Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2004-255

Integrating air transport elicits the need to harmonise software certification while maintaining safety and achieving security

E. Kesseler

This report has been based on an article published in the Aerospace Science & Technology Journal, Volume 8, Issue 4, Pages 347-358, June 2004 by Elsevier.

Customer:	National Aerospace Laboratory NLR
Working Plan number:	AV.1.J.4
Owner:	National Aerospace Laboratory NLR
Division:	Aerospace Vehicles
Distribution:	Unlimited
Classification title:	Unclassified
	June 2004

Approved by author:	Approved by project manager:	Approved by project managing department:
2/7/2004		02/07/2004

Contents

(14 pages in total)

Integrating air transport elicits the need to harmonise software certification while maintaining safety and achieving security

Ernst Kesseler [1]

NLR, Embedded Systems Department, A. Fokkerweg 2, 1059 CM Amsterdam, The Netherlands

Received 7 August 2003; received in revised form 4 March 2004; accepted 8 March 2004

Available online 15 April 2004

Abstract

Both Europe and the US have set ambitious new goals to improve air transport by simultaneously increasing capacity, reducing cost while improving an already impressive safety record. This requires integration of the systems of the various actors involved. The virtual enterprise concept, supported by a network-centric architecture, offers one possible solution. A prototype demonstrates the technical feasibility of this approach. Work on a certifiable safety-critical Java subset, the language used to implement the prototype, demonstrates the technical feasibility for each required safety level.

Unfortunately, current software certification standards differ for the various systems involved, imposing different and sometimes even non-compatible requirements. Based on the certification requirements of the prototyped services the applicable software certification standards are assessed. Network-centric solutions are based on the extensive use of Commercial-Off-The-Shelf (COTS) products and services. COTS is predicated on multiple users for a product or service, so the relevance of software certification schemes from other safety-conscious domains for air transport is reviewed to arrive at recommendations to improve the software certification process.

Without special provisions network-centric systems could lead to a new type of security vulnerability. Two remedial approaches, security certification and COTS security solutions are discussed below.

Keywords: Software safety certification; DO-178B; Software security; Common criteria; Virtual enterprise

1. Introduction

The European Vision 2020 [1] envisages in 2020 a threefold increase of air transport because air travel is more affordable, safer (five fold reduction of accident rate), very punctual (99% on time) and responsive (halve time-to-market). The vision explicitly mentions a very efficient air traffic management system and the need to combine the effort of all stakeholders e.g. through e-business, one of the constituents of the virtual enterprise, also referred to as virtual organisation.

The US Anyone, anything, anywhere, anytime vision [21] for 2025 states a 90% reduction of the fatal accident rate, a tripling of the air traffic management capacity and a reduction of the time-to-market from the current decades/years to months/weeks. Information technology heads the list of breakthrough capabilities required to achieve the ambitious US vision. General domain information technology can integrate the systems of the various stakeholders involved into a network-centric system-of-systems, a realisation of the virtual enterprise. Such a virtual enterprise improves the combined performance of all stakeholders involved. In various other domains, which lack air transport's safety concerns, such improvements have already been achieved. The Total Information Sharing for Pilot Situational Awareness Enhanced by Intelligent Systems (TALIS) project [17], which predates these visions but shares the cited conclusions, realises a prototype of a such network-centric architecture. This prototype enables a virtual organisation for the operational part of air transport. The completed prototype, consisting of the middleware and two sample applications, demonstrates the technical feasibility of this network-centric approach. Fig. 1 provides a conceptual overview of network-centric architecture.

The network-centric architecture supports services for all flight phases. Fig. 1 demonstrates the integration of various

E-mail address: kesseler@nlr.nl (E. Kesseler).
[1] Tel +31 20 511 3462. Fax +31 20 511 3210.

1270-9638/$ – see front matter
doi:10.1016/j.ast.2004.03.002

Nomenclature

AL	Assurance Level
ALARP	As Low As Reasonably Practicable
AOC	Airline Operations Centre
ATM	Air Traffic Management
CNS	Communication Navigation Surveillance
COTS	Commercial-Off-The-Shelf
EAL	Evaluation Assurance Level
EGNOS	European Geostationary Navigation Overlay System
EUROCONTROL	European Organisation for the Safety of Air Navigation
ESARR	EURCONTROL Safety Regulatory Requirement
FAA	Federal Aviation Administration
FHA	Functional Hazard Analysis
J2EE	Java Enterprise Edition
J2ME	Java Micro Edition
J2SE	Java Standard Edition
MISRA	Motor Industry Software Reliability Association
(P)SSA	(Preliminary) System Safety Assessment
SIL	Safety Integrity Level
TALIS	Total Information Sharing for Pilot Situational Awareness Enhanced by Intelligent Systems
WAAS	Wide Area Augmentation System

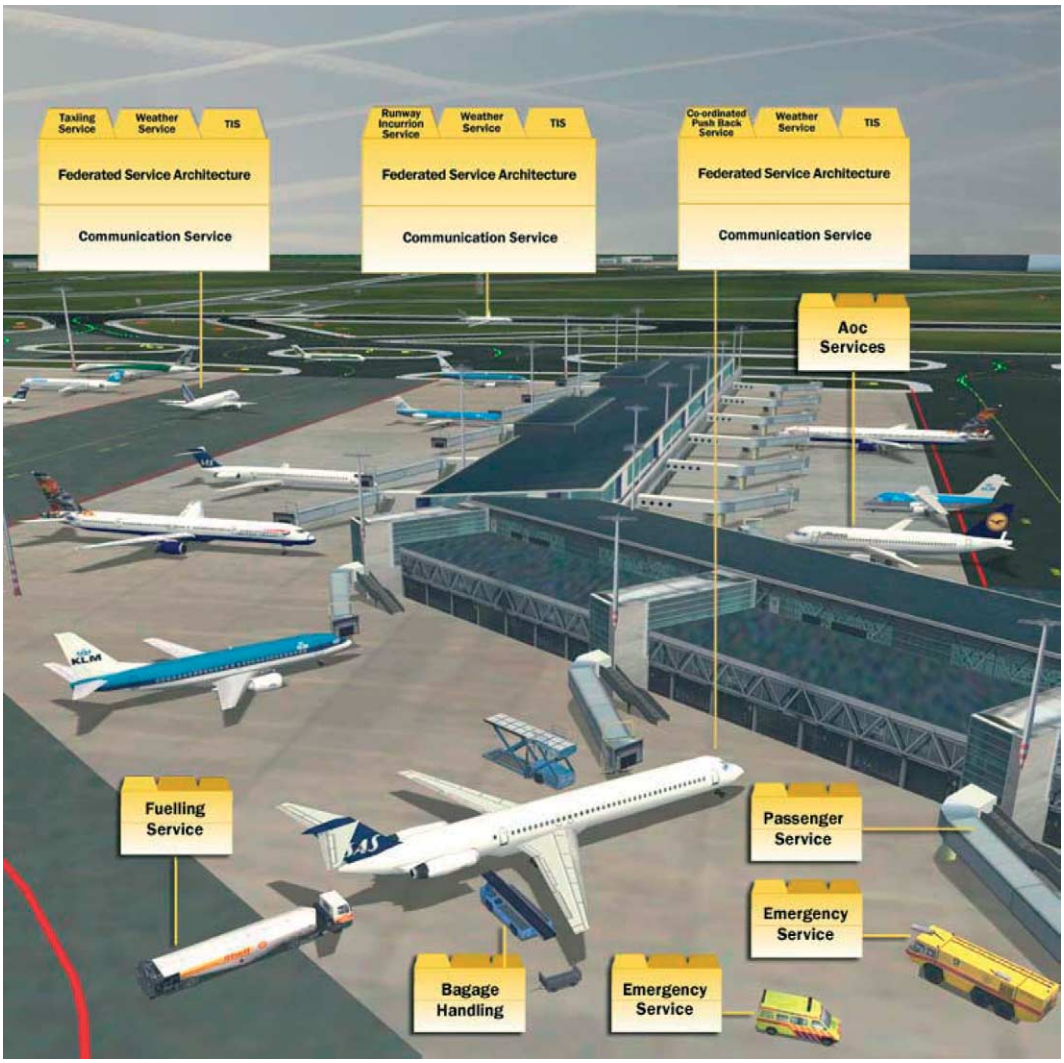

Fig. 1. Conceptual overview of the prototyped network-centric architecture.

actors at the airport. This part of the flight requires services of various safety criticality levels, supporting the need for harmonisation of the certification standards.

The pilot-oriented sample service of Fig. 2 illustrates the kind of optimisation that the network-centric architecture aims to support.

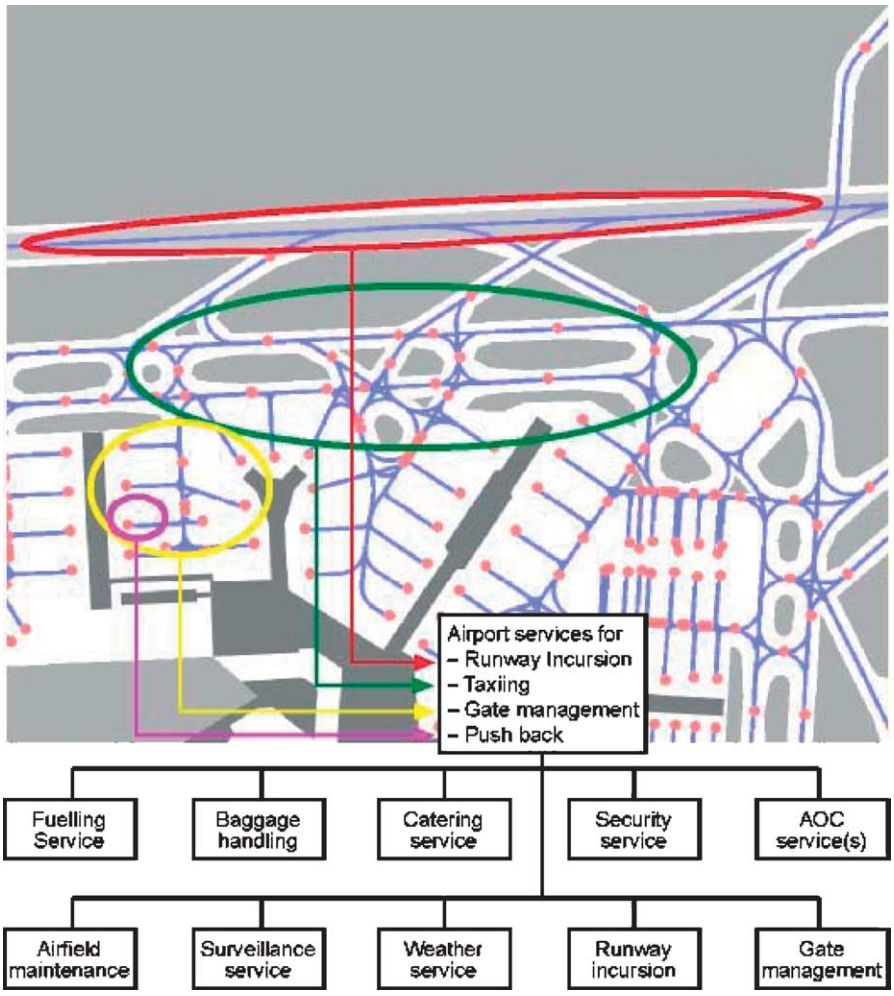

Fig. 2. Pilot-oriented sample service.

At an airport the pilot information-needs are flight-phase dependent. A co-ordinated pushback service will allow the pilot to improve the reliability of on-time pushback. For this the pilot needs amalgamated information from, e.g., fuelling services, baggage-handling services, catering services, security services and Airline Operations Centre (AOC) about transfer passengers. This pushback service optimises utilisation of the taxi-way linking the various gates and prevents aircraft from blocking each other or ending up in the wrong take-off order. Subsequently taxiing-services guide the aircraft to the correct runway, optimised for the other airfield traffic, its departure timeslot and taking possibly adverse weather or airfield maintenance restrictions into account. Finally, runway incursion services, using surveillance services, improve the safety during take-off.

Fig. 3 shows how the prototyped network-centric architecture builds upon various COTS components, which in turn support various hardware platforms, from small mobile wireless equipment (Java Micro Edition, J2ME) to standard PC-based hardware with standard communication (Java Enterprise Edition, J2EE). The services, which provide the actual value to the users, will run on top of the network-centric federated architecture. Once a service is connected to the federated architecture, all other services in the network can connect to it, either to provide input or use the results.

For the various systems and services depicted in Figs. 1 and 2, diverse certification standards and even different certification paradigms apply. The prototype and its intended services are used to focus which software certification standards to assess. Certifying the same software being part of several services according to different standards does not add safety but does impede the affordability and responsiveness required by the European and US vision statements.

Aircraft certification includes initial airworthiness certification of the aircraft type complemented by certification of each modification and upgrade of the aircraft. In principle certification is performed only once for each aircraft type. Pilots, other relevant airline personnel, aircraft maintenance personnel, etc. are certified or licensed for a fixed period of time. Extensions require re-certification effort. All certificates are issued and valid nationally, although the European Union is moving to a single certification for all its

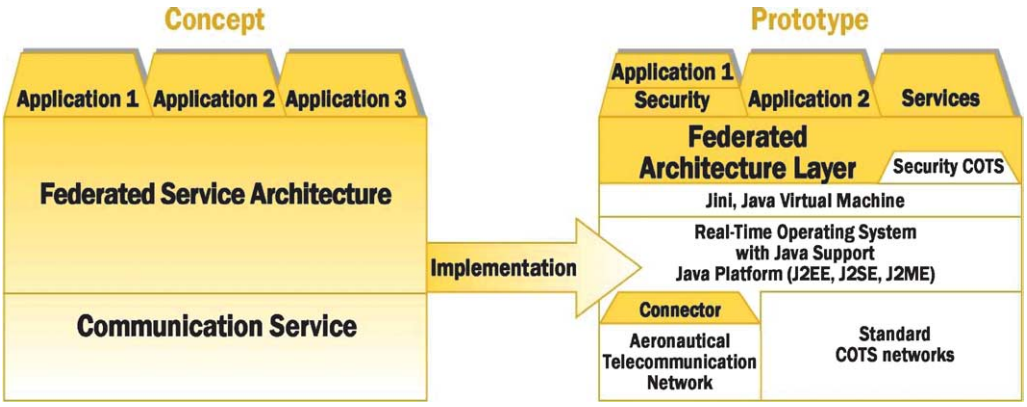

Fig. 3. Network-centric architecture overview.

member states. US-European harmonisation, which started in 1992, still has a long way to go to achieve a single certificate or a full, mutual recognition of the certificates. Depending on the country involved, these certifications are complemented by certification of the air traffic controllers and the air traffic management system. To ensure compliance with internationally recognised minimum safety standards for aircraft and pilots, international oversight of foreign aircraft and their pilots is being performed both in Europe and the United States. Over the years these certification systems have evolved into a comprehensive set of standards, rules and regulations, covering all relevant aspects of flying. However, as the various certification standards evolved from different backgrounds, they impose non-harmonised requirements. Components of network-centric systems will be software based. Relevant certification standards include DO-178B [4] for aircraft avionics, DO-278 [5] for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) systems, European Organisation for the Safety of Air Navigation (EUROCONTROL)'s Recommendation for Air Navigation Services software [8] and AC120-76A [9] for the electronic flight bag. Software certification can take a considerable amount of time and consume significant resources especially for systems designed without certification in mind as is demonstrated by the protracted and problematic Wide Area Augmentation System (WAAS) certification.

The realised prototype is based on using COTS and only adding the air transport specific functions. For most current software certification standards, use of COTS will exacerbate these certification problems. However the success of any network-centric solution critically depends on a paradigm shift to using COTS and achieving a short time-to-market. COTS can also help in achieving safety, as within the Open Group [20] work has started to produce a Java subset that will be certifiable to the strictest DO-178B level.

The major conclusion of [3] is that certification should assess the entire system and not be limited to the subsystems involved. Due to the importance of information technology to fulfill with the European and US visions, this paper will concentrate on software certification. Based on our own work on software certification, the relevant air transport practises will be assessed in the next section. Section 3 compares those practices with what is being done in some other safety-conscious sectors. Some potential security issues and solutions are addressed in Section 4. The last section arrives at recommendations on how to improve the current state-of-the-art to arrive at certifiable safe and secure but also responsive and affordable integrated air transport software.

2. Air transport software certification

2.1. General

All discussed standards share the notion that software has to be classified according to the system hazards (loss of life, aircraft damage) the software failure would cause or contribute to. This information is obtained from the Functional Hazard Analysis (FHA) combined with the (Preliminary) System Safety Assessment (P)SSA. Based on this information, the software will be classified. For each software class a number of standard specific requirements have to be satisfied. The current information technology state-of-the-art makes it impossible to assess whether a completed software product complies with the very low failure rates air transport requires. Consequently all standards impose requirements on the processes to make and maintain the software. An independent authority, i.e. independent from the supplier, purchaser and user, checks compliance with these process requirements and approves complying products as fit-for-use.

It is important that software certification focuses only on assuring sufficient confidence in achieving the required safety level. Safety certification should not interfere with the many other aspects related to software production, in order to allow the supplier maximum freedom to achieve its other organisational goals. Providing freedom on these other aspects will facilitate innovation and foster competition. Consequently, worthwhile goals like software improvement or ISO-9000 based quality assurance should be dealt with otherwise. EUROCONTROL acknowledges this principle of separation of concerns by dividing its regulatory

framework into mandatory rules, mandatory regulations and non-mandatory practises. For certain subjects these documents can be complemented by EUROCONTROL specifications, which provide a possible of means compliance, and guidelines, which provide explanation. The disadvantages of ignoring such separation of concerns are nicely illustrated by work on the EUROCONTROL's Air Navigation Services software recommendations (Section 2.4) which combine software safety and process improvement. After work started, ISO-12207, which describes the state-of-the-art in software processes, has been extended to ISO-15504. ISO-15504 defines 35 primary processes and 38 supporting and organisational lifecycle processes. ISO-9000 has been upgraded to ISO-9000: 2000 and the Capability Maturity Model has been upgraded to CMMIntegrated. All these standards are useful, but do not directly aim to assure software safety.

2.2. Airborne software safety standard DO-178B/ED12B

For all software in an aircraft DO-178B applies. As one of the oldest software safety standards it influenced other software safety standards. Based on the system level FAR/JAR AC-25-1309 five software levels are defined by DO-178B. For convenience in Table 1 the quantified FAR/JAR failure-probability definition is included.

Detailed requirements are provided for each level. Many consider DO-178B as the toughest standard in industry.

Fig. 4 provides an overview of the DO-178B software development and certification processes. DO-178B has an abstract lifecycle defining four generic phases (software requirement process, software design process, software coding process and software integration process). A developer must map its software processes onto those required by DO-178B. This is described in a special document called the Plan of Software Aspects of Certification. This document should be negotiated between the developer and certifying authority prior to actual software development. Subsequently, the developer needs only to comply with the agreed Plan of Software Aspects of Certification.

DO-178B specifies 66 detailed process requirements. For every level the applicability of each requirements is defined, with all required for level A.

Industry standards define the functions that must exist in certain avionics units (e.g. flight management system). Suppliers can further enhance such units with customer-based requirements. Consequently, most avionics units are custom-made, though the mandating of core functionality provides a basis for reuse, i.e. COTS.

To date, there is usually only one software application assigned to a hardware unit. The integrated modular avionics industry standard will allow several fixed and pre-defined applications to run on the same hardware unit. This exam-

Table 1
DO-178B/ED12B overview

Level	System failure	Failure description	Failure probability description	FAR/JAR definition per flight hour
A	Catastrophic failure	Aircraft loss and/or fatalities	Extremely improbable	$\cdots < 10^{-9}$
B	Hazardous/severe major	Flight crew can not perform their tasks Serious or fatal injuries to some occupants	Extremely remote	$10^{-9} < \cdots < 10^{-7}$
C	Major failure	Workload impairs flight crew efficiency Occupant discomfort including injuries	Remote	$10^{-7} < \cdots < 10^{-5}$
D	Minor failure	Workload within flight crew capabilities Some inconvenience to occupants	Probable	$10^{-5} < \cdots$
E	No effect	No effect	Not applicable	–

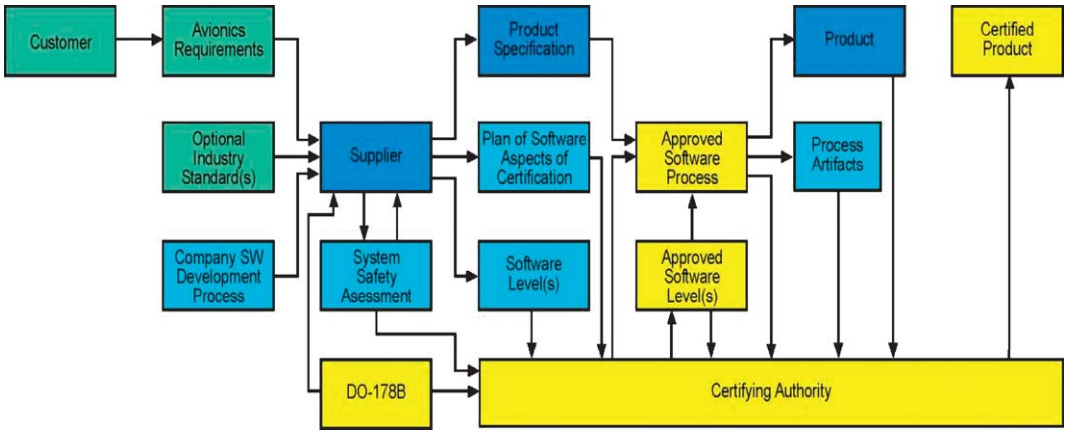

Fig. 4. Overview of airborne safety standard DO-178B processes.

ple illustrates how DO-178B trails current real-time and embedded systems development practices where multitasking is common practise. The first applicant for a new technology bears the full burden of convincing the certifying authority.

Commercial off-the-shelf (COTS) products are officially allowed by DO-178B, however no requirements are waived for COTS that have been certified to other safety-standards. Consequently, only COTS products that have been developed specifically taking all DO-178B requirements into account can be used. Note that navigation services like European Geostationary Navigation Overlay System (EGNOS) and the European Galileo effort, all take DO-178B into account but deviate on details in project specific certification documents. The EGNOS software engineering standard DRD920 [6] contains a Plan of Software Aspects of Certification stating full compliance with forty DO-178B requirements plus partial compliance for another four. These deviations are predicated upon cost-benefit analyses. Some requirements may be more expensive than justified by air transport's the relative usage of the EGNOS service. Each deviation must be shown to the conservative certifying authority not to materially distract from the safety goals of DO-178B or be met, implying additional certification effort and consequently increasing the time-to-market and costs.

The safety appeal of air transport for other safety-conscious markets has led some operating systems vendors to produce kernels, which together with the application have successfully passed DO-178B certification. The vendor will provide all artefacts needed to certify a new application reusing the same kernel. This reduces the certification effort to only the new application being developed. All these kernels allow software of different safety criticality levels to run concurrently on the same hardware. One product is compatible with ARINC 653, also referred to as integrated modular avionics. Such COTS will reduce costs and provide additional capabilities like multitasking that air transport alone can not afford.

Certification is required from each nation where an airline wants to acquire an aircraft for civil use. Airbus has obtained its initial 13 type certifications over the last 10 years from the European Joint Aviation Authority, complemented by another 13 from the U.S. Federal Aviation Administration (FAA) plus 130 from other nations. Boeing obtained 200 additional certificates in the same period, after the initial FAA certification [11]. Substantial benefits can be accrued when each nation accepts the certifications of all accredited ICAO member states. A system of accreditation should ensure equal enforcement of the standard in all nations concerned to prevent safety degradation occurring in specific nations to propagate internationally, the unwanted equivalent of the maritime cheap flag countries. Currently mutual certification recognition involves a lengthy negotiation between the two certifying authorities involved, leading to a bilateral agreement.

Air transport's good safety record does not repudiate the claim that DO-178B compliance provides the safety objectives. However, catastrophic failures (level A) are fortunately so rare, that the absence of software-induced catastrophic failures does not statistically justify DO-178B claims. Like all other software safety standards, evidence on the necessity and effectiveness of each of the 66 requirements is lacking. They are based on a consensus on engineering judgement.

Table 2
DO-278/ED109 overview

DO-178 level	DO-278 Assurance Level	COTS service experience
A	AL 1	Not allowed
B	AL 2	Negotiate with approval authority
C	AL 3	One year
	AL 4	Six months
D	AL 5	Typically not needed
E	AL 6	Not applicable

2.3. RTCA/EUROCAE Air Traffic Management standard DO-278/ED109

For Air Traffic Management (ATM) ground and satellite systems, the RTCA/EUROCAE have produced a new standard DO-278 by extending DO-178B. Table 2 provides an overview of the six Assurance Levels (AL) defined in DO-278. Note that unlike DO-178B, neither a definition of the assurance levels nor an indication of the allowed failure probability is provided. DO-278 adds an assurance level by splitting level C. DO-178B added a level by splitting level II from it predecessor DO-178A into level B and C. Consequently any software safety standards should provide sufficient grading.

In contrast to DO-178B, DO-278 acknowledges the use of independently developed (pre-existing) COTS, by defining processes for planning, acquisition, verification, configuration management and quality assurance. It must be demonstrated that unused COTS capabilities do not adversely effect the ATM system. An important extension to DO-178B is that COTS service experience may be used, thereby obviating the need to apply a DO-278 compliant development process for some assurance levels. However, the restrictions on service experience are quite severe. The information on service experience is included in Table 2. In the table, "one year" means that for a continuous period of 8760 hours of representative use no failure may occur. Additionally, all in-service reports originating from all users of the COTS have to be evaluated for their potential adverse effects on the ATM system.

2.4. EUROCONTROL Air Navigation Services standard

EUROCONTROL has produced Recommendations for Air Navigation Services Software which are an elaboration of Software in ATM Systems, EUROCONTROL Safety Regulatory Requirement ESARR6. ESARR6 is currently subject to a formal approval process and consequently not yet publicly available. Its software safety classification is

Table 3
EUROCONTROL Air Navigation Services software recommendations overview

ANS software assurance level	ESARR4 severity (class, effect)		ESARR4 likelihood of occurrence	Software likelihood of occurrence (per operational-hour)
1A	1	Accidents	Improbable	N/A
1B		DO-178B level B	N/A	DO-178B Extremely Remote $10^{-9} < \cdots < 10^{-7}$
2	2	Serious incidents	Remote	$10^{-6} < \cdots < 10^{-5}$
3	3	Major incidents	Occasional	$10^{-5} < \cdots < 10^{-4}$
4	4	Significant incidents	Probable	$10^{-4} < \cdots < 10^{-3}$
5	5	No immediate effect on safety	N/A	N/A

based on the ESARR4 [7]. This standard combines DO-178B, IEC-61508, and the capability maturity model into a combined safety and quality assurance document. The software classification provided in Table 3 is based on ESARR4, while inserting an additional level identical to level B of DO-178B. The requirements on the evidence that needs to be provided depend on the assurance level.

Useful innovations are guidance for contracting software, both for acquisition and for supply plus guidance for operations and maintenance. Guidance for COTS is provided. For assurance levels 1A and 1B COTS is not allowed.

2.5. Electronic flight bag AC120-76A

The electronic flight bag is a COTS-based hardware platform that supports many independent software applications, even simultaneously. As such the electronic flight bag is well suited for the airborne part of network-centric solutions. The electronic flight bag can be part of the aircraft and so DO-178B applies. However, it can also be used outside the aircraft, so a special document on the safety and certification AC120-76A is available. The electronic flight bag could either be a portable device like a slate laptop or personal digital assistant, or be installed in the aircraft.

The electronic flight bag software is classified as:

- Type A: applications that include pre-composed, fixed presentations of aviation data. Type A software needs Flight Standards District Office approval. 71 example applications are provided.
- Type B: applications that include dynamic applications that interactively manipulate and present aviation data. Type B software additionally needs evaluation by the Aircraft Evaluation Group. AC120-76A lists 17 applications. A six-month operational evaluation is needed, during which a (paper) back up of the application is required.
- Type C: all other applications. Full DO-178B approval is needed.

All user-modifiable software is type C. Consequently a key requirement of network-centric solutions like dynamically uploading applications remains cumbersome. Positive is the guidance provided on usability and on human factors.

Compliance to AC120-76A implies compliance to 103 sections of 5 parts of the US Code of Federal Regulation relating to airworthiness plus 45 additional sections of 4 parts of the operating regulations plus 20 advisory circulars plus 10 other FAA regulations. Even within AC120-76A, some parts relate to activities performed only once for the approval of software, while other parts mention an operational approval valid for a specific operator for six months. This profusion of standards, regulations, etc., is typical for network-centric integrated systems. [21] bluntly states that current certifications processes can not handle the integrated software based systems which are needed to fulfill its vision. However neither research nor experience supports their proposed shift from product certification at subsystem level to process certification at system level. As observed by [3], even the current, more limited, delegation of certification responsibility to suppliers, through the designated engineering representative, resulted in (occasional) approval of deficient or non-complying products. A better option could be to improve various existing data reporting mechanisms to obtain evidence on the effectiveness of the software-safety requirements, which is statistically feasible for the lower criticality levels. Consequently, there is a need for a different approach to certification, which ensures the safety, but omits the many not harmonised standards. Note that possible different national interpretations of certification requirements complicate certification without increasing safety.

3. Other domain safety standards

The following overview from software safety standards from other domains is provided:

- To learn from their approach. Consequently this chapter will focus on the differences.
- To assess COTS. For COTS to be viable the air transport market is too small. Both Boeing and Airbus only produce a few hundred aircraft per year, with the number of processors involved in safety critical tasks in the tens per aircraft. The annual number of new ATM systems is much smaller. Consequently, COTS products should be deployable in other safety-conscious markets as well.

Table 4
IEC-61508 Safety Integrity Levels overview

SIL	Failure probability per hour (systems active > once per year)	Failure probability per demand (systems active < once per year)
4	$10^{-9} \leqslant \cdots < 10^{-8}$	$10^{-5} \leqslant \cdots < 10^{-4}$
3	$10^{-8} \leqslant \cdots < 10^{-7}$	$10^{-4} \leqslant \cdots < 10^{-3}$
2	$10^{-7} \leqslant \cdots < 10^{-6}$	$10^{-3} \leqslant \cdots < 10^{-2}$
1	$10^{-6} \leqslant \cdots < 10^{-5}$	$10^{-2} \leqslant \cdots < 10^{-1}$

3.1. Process industry IEC-61508

From the general software safety-critical domain [14] is available which originated in the process industry. Four Safety Integrity Levels (SIL) are defined. IEC-61508 states that only for hardware, safety can be quantified and assessed using reliability prediction techniques. For software, only qualitative techniques and judgements are possible. The Standard explicitly states that failures rates lower then 10^{-9} per hour (i.e., level A according to DO-178B or DO-278 AL1) can not be achieved for software. IEC-61508 defines its own software safety lifecycle, based on the V-model. The lifecycle includes operations, maintenance, repair, retrofit and even decommissioning procedures, an extension to DO-178B. The railway industry is converting to IEC-61508 by providing domain specific extensions to IEC-61508, in line with the intention of this standard.

Part 7 of the Standard aims to provide an exhaustive list of techniques for each process phase, including recommendations on their use (or avoidance) for each SIL. This part of the standard will need regular updates to remain in-line with information technology innovations. In accordance with the general IEC practise the first update is due in 2006. It is possible to certify COTS for a certain level, when an IEC-61508 compliant development process is followed. An independent party will perform the certification. Table 4 provides an overview of the four SIL levels.

Using service experience is allowed, but in practice hardly possible for higher SIL levels. An example from the standard states that for a SIL 1 system, 95% confidence in correct functioning requires 300 hours of relevant service experience. For a SIL 4 system, 99.5% confidence requires 690 000 years of service experience.

3.2. Nuclear industry IEC-60880-2

In the nuclear industry IEC-60880-2 [12] is applicable. IEC-60880 is based on the software classification provided in IEC-61226 [13], see Table 5. This industry's basic single-failure criterion requires the assembly of safety systems to remain functional despite any random failure. This single-failure criterion is not applicable for software, as a software failure can cause a system with multiple hardware units to fail. As a consequence IEC-60880 devotes an appendix to the pros and cons of multiple diverse software implementations. Multiple diverse software versions only provide protection against some fault classes, so incorrect or ambiguous specifications remain a single point-of-failure.

Table 5
IEC-61226 overview

Category	Description	Excerpt assignment criteria
A	Principal role in achieving safety	• Mitigate to prevent significant sequence • Failure could result in significant sequence
B	Complementary role to category A	• Control process variables within safety limits • Alert staff of category A failure • Continuously monitor category A function
C	Auxiliary or indirect role	• Enhance category A performance • Monitor and mitigate internal hazards and natural events • Ensure personnel safety
Unclassified	No direct safety role	• Not significant to safety

IEC-60880-2 distinguishes between software tools that can introduce errors and tools that fail to detect them. The requirements for the former category are strict. Compilers (called translators) are acknowledged to be too large to demonstrate their correctness. They are trusted under certain restrictions, unlike DO-178B where binary code needs to be verified for the highest level. The compiler may not introduce dead code, which is code that is not traceable to requirements (e.g., error handling). Operating experience may compensate for some lack of design documentation.

IEC-60880-2 allows COTS. There are strict requirements on the evaluation of functions, design documentation etc. In case operating experience is used, there are requirements on the operating history data. Like DO-278 after acceptance of the COTS, all subsequent error and failure information from all users has to be assessed for its potential impact on the approved system.

Formal methods can prove with mathematical rigour that an implementation conforms to its specification. Each of the wide range of formal methods seems to work best for certain specific applications. Unfortunately formal methods in general do not yet scale well to applications with the size of actual safety critical applications. For tractable systems formal methods have shown to provide benefits. For instance 70% of the A340 flight control computer software can be generated using a specific formal method plus automatic code generation tool [2]. This application is of the highest criticality level. Unlike DO-178B, the evidence provided by formal methods, where available, is deservedly recognised by IEC-60880-2.

3.3. Medical industry FDA-1252

For software in medical devices in the USA [10] applies. The software is classified into three "levels of concern", see Table 6. FDA-1252 states that as the probability of software failure cannot be measured, only the severity of the software failure consequences is used to determine the level of concern. A table listing 12 documents describes for

Table 6
FDA-1252 Level of concern

Level of concern	Severity description
Major	Software failures that could cause, directly or indirectly, to death or serious injury of the patient and/or the operator
Moderate	Software failures that could cause, directly or indirectly, to non-serious injury of the patient and/or the operator
Minor	Software failures are not expected to cause injury to patient and/or operator

Table 7
MISRA overview

Controllability category	Acceptable failure rate	Occurrences per year	Integrity level
Uncontrollable	Extremely improbable	10^{-3}	4
Difficult to control	Very remote	10^{-2}	3
Debilitating	Remote	10^{-1}	2
Distracting	Unlikely	1	1
Nuisance only	Reasonably possible	10	0

each level of concern what type of information is needed, if any. No specific software life-cycle model is prescribed, but a general V-model for verification is provided. Verification needs to be performed at module, integration and system levels.

Even though FDA-1252 states that artificial neural networks are impossible to verify, they are allowed for all levels of concern. Consequently the assumptions and the training of the neural network need to be verified, but no guidance is provided.

According to FDA-1252, embedded and real-time systems pose unique concerns. The use of techniques, simulators and emulators to analyse timing of critical events is only mentioned, but not imposed. The importance of human factors is acknowledged without enforcing verification and validation requirements. In the same spirit, security is raised, but no requirements ensue. Consequently the air transport domain can not learn much from FDA-1252. However, in order for COTS to become commercially viable, COTS needs to be deployable in various safety critical domains. This implies recognition of DO-178B by the medical domain and a scaling of its levels to the FDA-1252 levels of concern.

3.4. Automotive industry, MISRA report 2

The Motor Industry Software Reliability Association (MISRA) has a very practical approach to software integrity in [18]. The document contains an informative example to determine the software integrity. Table 7 contains their classification and an indicative occurrence level.

For integrity level 4 MISRA recommends the use of formal methods complemented with automatic code generation, although with current technology this is not possible. The use of a certified compiler of a language with formal semantics is recommended. Until they become available, like DO-178B, it has to be shown that the machine code reflects the high-level language version. The assessor needs unimpeded access to all software development information, something implicitly assumed by DO-178B.

3.5. ISO 15026 software integrity levels

The ISO 15026 [15] standard combines a threat-identification with a frequency analysis and a consequence analysis to determine the software integrity level. The occurrence frequency varies between probable, which means 1 to 10^{-1} occurrences per year and incredible with less then 10^{-6} occurrences per year, which is low with respect to other standards. The severity of consequence has four grades from catastrophic via major and severe to minor. Using the ALARP (As Low As Reasonably Practical) principle four software integrity levels are determined, A (high), B (intermediate), C (low) and D (trivial). The methods and checks to achieve sufficient confidence for the allocated software integrity level are outside the scope of the standard. Such methods and checks have to be agreed upon on a national, sector specific or project specific basis. The advantage of the minimalist approach is that it is more likely to remain stable over the years and that it is well suited for many application domains. The disadvantage is that additional standards or guidance are needed to determine the amount of confidence in achieving the software integrity level.

4. Security

4.1. General

After the tragic September 11 (2001) events new security threats have become a major concern for air transport. At the moment only a very limited number of computer networks is used in those parts of the air transport system which have safety concerns. Humans using voice links do most communication. Currently air transport specific proprietary technology is used, which provides a level of protection against the ubiquitous hackers that unfortunately roam the general domain. Especially open COTS-based network-centric systems, like the realised prototype, are vulnerable to this type of attack, which is new to air transport, hence protection is needed. The trend to network-enable systems will also become relevant for the safety-critical systems in the other safety-conscious domains. This implies that standards are not only needed for safety but could also be used to ensure security. The disadvantage of adding security functions to each networked service is a proliferation of potentially un-harmonised security services.

Another approach is to deploy security middleware and rely on general domain COTS to provide and maintain a sufficient level of security. The resulting separation of concerns allows resources to be concentrated on solving air

transport specific issues. An example of each approach is described below.

4.2. ISO-15408

ISO-15408 [16] is an international standard that includes security requirements and can provide certifications for complying products. ISO-15408 is the civil variant of the Common Criteria from the military domain and will follow all updates of the military Common Criteria.

ISO-15408 provides objective evidence about the product security level. Qualified and officially recognised assessors perform the objective and repeatable evaluation, much like DO-178B for safety certification. The evaluation can lead to a certificate, which is currently recognised by 8 countries plus all (15) European Union countries.

The ISO-15408 considers three security objectives aiming to prevent:

- Damaging disclosure of the service to unauthorised recipients (loss of confidentiality).
- Damage through unauthorised modification (loss of integrity).
- Damage through unauthorised deprivation of access to the asset (loss of availability).

Fig. 5 provides an overview of the ISO-15408 view on the realisation of security functions. The security environment provides the context of the asset. Combined with the perceived threats and the security policy the security requirements can be derived. These requirements consist of a reusable Protection Profile and an asset specific Security Target. Based on these requirements and the extensive listing of possible security functions in the ISO-15408 Part 2, the security functions of the system are determined. Separately the protection level is determined, which determines the amount of implementation effort and evaluation effort. Table 8 provides an overview of the Evaluation Assurance Levels (EAL's). The amount of COTS products and reusable protection profiles in the register at the time of writing (January 2004) illustrates that ISO-15408 is rapidly being accepted. Note that if a product fully complies with the EAL requirements but additionally is certified to comply with some requirements of the next higher EAL this can be denoted by N+ in the certificate and consequently in Table 8 which is based on the register.

ISO-15408 adds further requirements on the software development process, which are based on consensus of engineering judgement. Harmonisation with the safety requirements is advantageous.

Note that whereas DO-178B unjustifiably does not recognise the verification evidence from formal methods, ISO-15408 requires it for the highest level. As air transport does not have a tradition in software security certification, the industry can benefit of the military and commercial domains through this more advanced standard.

4.3. Security middleware

The disadvantage of ISO-15408 is that every application or service needs to implement the security requirements itself. As all intended services are network-enabled, it is possible to deploy security middleware, just like the realised prototype deploys network-centric middleware. Such security middleware takes care of the authentication, an important security aspect. Once a user is authenticated, personalised access to all authorised services can be granted e.g. only the fuelling-service for the fuelling personnel and many services for the pilot in Fig. 1. The advantage of a middleware solution is that the services themselves do not need to implement the security. Suitable middleware, e.g. A-Select [19], allows using various authentication mechanisms concurrently and transparently like a light authentication for the fuelling personnel with a strong authentication for the pilot. Middleware facilitates a quick deployment with existing, low strength mechanisms, with an option to upgrade to

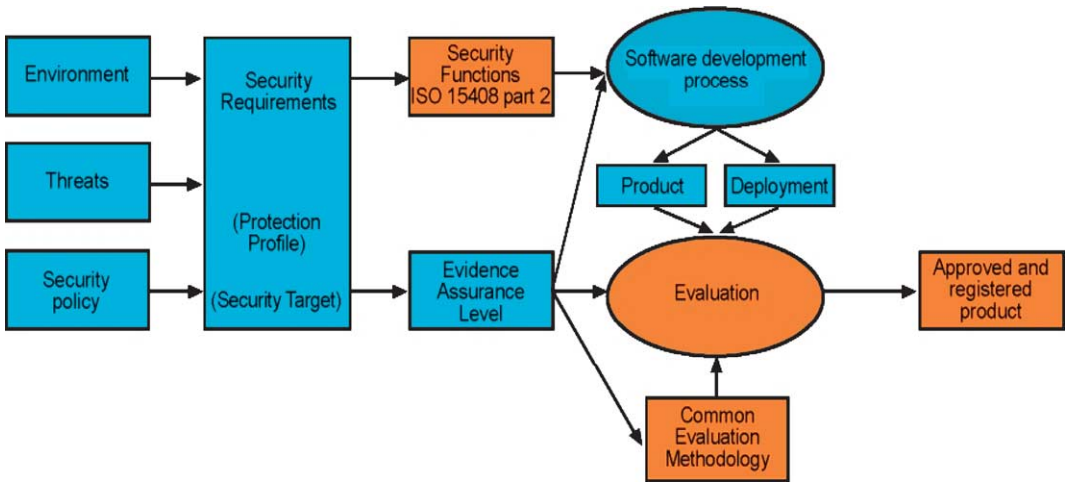

Fig. 5. Overview of ISO-15408 security standard.

Table 8
ISO-15408 Evaluation Assurance Level

EAL	Description		# of COTS products		# of protection profiles	
			Certified	In evaluation	Certified	In evaluation
1	Functionally tested,		10	0	0	0
	security threats not serious	1+	1			
2	Structurally tested,		18	39	1	7
	low to moderate assurance	2+	9	4		
3	Methodically tested and checked,		14	11	4	4
	maximum assurance without infringing sound development practise	3+	3	6		
4	Methodically designed, tested and reviewed,		30	4	8	3
	maximum assurance compatible with good commercial practise	4+	13	7		
5	Semiformally designed and tested,		0	0	0	1
	maximum assurance with moderate security engineering	5+		1		
6	Semiformally verified design and tested, protect high value assets against significant risk		0	0	0	0
7	Formally verified design and tested, extremely high risk situations and/or high assets values		0	1	0	0
	Total # of COTS products		95	34	13	15

stronger certified mechanisms once they become available. The ability to use COTS security functions for requirements that were not included in the original demonstrator shows the benefits accrued from using the realised open architecture.

5. Conclusions

The European and US visions for air transport require integration of the systems of the various actors involved. Prototypes based on the deployment of COTS, like TALIS, show that such integration can be achieved. Current work on certifiable Java will technically allow compliance with the most stringent safety requirements. However, current air transport certification standards and practises, like DO-178B, do not recognise any certification evidence which is gathered outside the rigorous air transport certification schemes. Current certification schemes are based on strict process requirements, without scientific or empirical backing. For software this has led to a profusion of standards with different or sometimes even non-compatible requirements. Consequently, it is difficult to recognise certification evidence from other domains. Those differences are hard to justify, as they derive from common safety concerns. No standard is clearly superior. For integrated systems that have to rely on COTS, like service-driven network-centric solutions, the current certification practise is hampering key requirements like responsiveness and affordability hence improvement is needed.

To improve the following attributes are recommended for inclusion in software safety standards and certification schemes:

- Certification by an independent third party.
- Trusted party or organisation must accredit the independent third party, yielding international recognition.
- Certification in accordance with objective standards.
- Objective guidance for the reviewers (e.g. like for DO-178B and ISO-15408).
- Maximum freedom for the software processes being used and the techniques being deployed in order to exploit information technology innovation.
- Grading of software with sufficient nuances for less safety critical applications.
- Data gathering to substantiate the existing software-safety requirements or modify them to effective requirements.
- Recognise the safety evidence of COTS products in a timely and cost effective manner, to preserve the advantages of deploying COTS.
- Mutual recognition of the various standards/certificates to facilitate re-use from safety-conscious domains like medical, nuclear, railway, process and automotive industry.
- The current state-of-the-art does not (yet) allow large systems (as used in air transport) to be formally verified. For tractable subsystems, the evidence provided by formal methods should be recognised.
- Small, focused scope, which facilitates re-use and improves responsiveness to changes in the continuously evolving information technology.
- Software standards covering various non related properties, like safety and security, should impose compatible requirements on the software lifecycle.

Research is needed on the effect of software safety requirements so each requirement can be justified for the intended safety level. Standard innovations like the goal-based approach, which states the objective, the evidence and the reasoning are beneficial. It may comply with the objective of the US aerospace committee to shift to a new software certification paradigm.

Security middleware separates authentication from the applications, facilitating a quick and affordable way to introduce security in air transport.

References

[1] P. Argüeles, et al., Report of the group of personalities, European aeronautics: a vision for 2020, http://europa.eu.int/comm/research/growth/aeronautics2020/pdf/aeronautics2020_en.pdf, January 2001.

[2] J.L. Camus, Efficient development of airborne software with Scade suite, http://www.esterel-technologies.com/v3/?id=41490#DO-178B, 2003.

[3] D.I. Chaney, et al., Commercial aeroplane certification process study, FAA, http://www.asme.org/gric/engineeringpolicy/Images/piscopo.pdf, March 2002.

[4] DO-178B/ED12B, Software Considerations in Airborne Systems and Equipment Certification, RTCA & EUROCAE, December 1992.

[5] DO-278/ED109, Guidelines for the communication, navigation surveillance, and air traffic management (CNS/ATM) systems software integrity assurance, RTCA & EUROCAE, March 2002.

[6] GNSS-1 Programme implementation phase, EGNOS software engineering standard, to be obtained from the EGNOS programme office, August 1999.

[7] ESARR4 Risk Assessment and Mitigation in ATM, EUROCONTROL, http://www.eurocontrol.be/src/html/deliverables.html, October 2002.

[8] EUROCONTROL Recommendations for ANS software, proposed issue March 2003.

[9] FAA AC120-76A, Guidelines for the certification, airworthiness and operational approval of electronic flight bag computing devices, FAA, July 2003.

[10] FDA-1252, Guidance for FDA reviewers and industry guidance for the content of pre-market submissions for software contained in medical devices, http://www.fda.gov/cdrh, May 1998.

[11] H. Holderbach, Type certification of commercial aircraft, call for enhanced international rules, ICAO J. 2 (2001).

[12] IEC-60880, Software for computers important to safety for nuclear power plants, Part 2, software aspects of defence against common cause failures, use of software tools and of pre-developed software, http://ww.iec.ch, December 2000.

[13] IEC-61266, Nuclear power plats – instrumentation and control systems important for safety – Classification, http://ww.iec.ch, May 1993.

[14] IEC-61508 Functional safety: safety related systems, 7 parts, http://ww.iec.ch, December 1998.

[15] ISO 15026 Information Technology – System and Software Integrity Levels, http://www.iso.ch, November 1998.

[16] ISO-15408 Common criteria for security evaluation, Version 2.1, also known as the Common Criteria, http://www.commoncriteria.org/cc/cc.html, August 1999.

[17] E. Kesseler, Transforming air transport to a concurrent enterprise: Technical, safety and security perspectives, ICE2003, June 2003.

[18] MISRA Report 2, Integrity, http://www.misra.org.uk/, February 1995.

[19] A. Odaci, Overview of A-Select, http://a-select.surfnet.nl/aselect_overview.html, August 2003.

[20] Open Group, Real-time and Embedded Systems Forum, Java Specification Request, http://www.opengroup.org/rtforum/ADD, August 2003.

[21] R.S. Walker, et al., Commission on the future of the US aerospace industry, Anyone, anything, anywhere, anytime, http://www.aerospacecommission.gov/AeroCommissionFinalReport.pdf, November 2002.